UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/648,801	10/10/2012	ROBERT R. FRIEDLANDER	END920120186US1	5087

79230 7590 01/18/2017

Law Office of Jim Boice
3839 Bee Cave Road
Suite 201
West Lake Hills, TX 78746

EXAMINER
PRINGLE-PARKER, JASON A

ART UNIT	PAPER NUMBER
2618	

NOTIFICATION DATE	DELIVERY MODE
01/18/2017	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

Jennifer@BoiceIP.com
Emily@BoiceIP.com
Jim@BoiceIP.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte ROBERT R. FRIEDLANDER, JAMES R. KRAEMER, and JEB R. LINTON

Appeal 2016-001990[1]
Application 13/648,801
Technology Center 2600

Before ALLEN R. MacDONALD, JOHN P. PINKERTON, and GARTH D. BAER, *Administrative Patent Judges*.

BAER, *Administrative Patent Judge*.

DECISION ON APPEAL

---

[1] Appellants identify International Business Machines Corporation as the real party in interest. Appeal Br. 2.

STATEMENT OF THE CASE

This is a decision on appeal, under 35 U.S.C. § 134(a), from the Examiner's Final Rejection of claims 1–20, which are all the pending claims. Appeal Br. 1. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm-in-part and enter a new ground of rejection under 37 C.F.R § 41.50(b).

BACKGROUND

A. The Invention

Appellants' invention is directed to "[managing] nodes in a graph database." Abstract. Claims 1, 2, 4, 7, 8, and 15 are representative and reproduced below, with emphasis added to the disputed elements:

> 1. A processor-implemented method of managing nodes in a non-hierarchical graph database, the processor-implemented method comprising:
>
> receiving, by the processor, a data stream that describes graph nodes in a non-hierarchical graph database;
>
> defining, by the processor, multiple graph node clusters from the graph nodes in the non-hierarchical graph database; and
>
> ***generating, by the processor, a cluster edge between two graph node clusters from the multiple graph node clusters in the non-hierarchical graph database, wherein the cluster edge describes a relationship between the two graph node clusters.***
>
> 2. The processor-implemented method of claim 1, further comprising:
>
> ***generating a display of the non-hierarchical graph database, wherein the non-hierarchical graph database comprises the two graph node clusters, the cluster edge, and details of the relationship between the two graph node clusters.***

4. The processor-implemented method of claim 1, wherein each of the graph nodes represents a synthetic context-based object, and wherein synthetic context-based objects are generated by:

associating, by the processor, a non-contextual data object with a context object to define a synthetic context-based object, wherein the non-contextual data object ambiguously relates to multiple subject-matters, and wherein the context object provides a context that identifies a specific subject-matter, from the multiple subject-matters, of the non-contextual data object.

7. The processor-implemented method of claim 1, ***wherein at least one of the two graph node clusters is empty, wherein an empty graph node cluster provides a structure for holding graph nodes at a future time.***

8. A computer program product for managing nodes in a non-hierarchical graph database, the computer program product comprising a tangible non-transitory computer readable storage medium having program code embodied therewith, the program code readable and executable by a processor to perform a method comprising:

receiving a data stream that describes graph nodes in a non-hierarchical graph database;

defining multiple graph node clusters from the graph nodes in the non-hierarchical graph database; and

generating a cluster edge between two graph node clusters from the multiple graph node clusters in the non-hierarchical graph database, wherein the cluster edge describes a relationship between the two graph node clusters, and wherein the relationship between the two graph node clusters comprises a description of an upstream connection from one of the graph node clusters to an upstream node

cluster, and wherein the relationship between the two graph node clusters comprises a description of a downstream connection from one of the graph node clusters to a downstream node cluster.

15. A computer system comprising:

a processor, a computer readable memory, and a computer readable storage medium;

first program instructions to receive a data stream that describes graph nodes in a non-hierarchical graph database;

second program instructions to define multiple graph node clusters from the graph nodes in the non-hierarchical graph database; and

third program instructions to generate a cluster edge between two graph node clusters from the multiple graph node clusters in the non-hierarchical graph database, wherein the cluster edge describes a relationship between the two graph node clusters;

fourth program instructions to associate a non-contextual data object with a context object to define a synthetic context-based object, wherein the non-contextual data object ambiguously relates to multiple subject-matters, and wherein the context object provides a context that identifies a specific subject-matter, from the multiple subject-matters, of the non-contextual data object;

fifth program instructions to combine synthetic context-based objects that each contain a same non-contextual data object and a different context object from a first set of different context objects into a first synthetic context-based objects graph node cluster;

sixth program instructions to combine synthetic context-based objects that each contain the same non-contextual data object and a different context object from a second set of different context objects into a second synthetic context-based objects graph node cluster; and

> seventh program instructions to display the same non-contextual data object in a context-based cluster edge that links the first synthetic context-based objects graph node cluster to the second synthetic context-based objects graph node cluster; and wherein the first, second, third, fourth, fifth, sixth, and seventh program instructions are stored on the computer readable storage medium for execution by the processor via the computer readable memory.

Appeal Br. 16–20 (Claims App.).

B. The Rejections on Appeal

The Examiner rejects claims 7 and 14 under 35 U.S.C. § 112, second paragraph as indefinite for failing to particularly point out and distinctly claim the subject matter that the applicant regards as the invention. Final Act. 5.

The Examiner rejects claims 1–20 under 35 U.S.C. § 102(b) as anticipated by Grigory Begelman et al., "Automated Tag Clustering: Improving Search and Exploration in the Tag Space," May 2006, https://docs.google.com/file/d/0B0uw1JCogWHuTFQzY3VKVEJMOGM/edit (last visited Jan. 5, 2017) ("Begelman").[2] Final Act. 6.

[2] A paper associated with the cited presentation is also found at Grigory Begelman et al., "Automated Tag Clustering: Improving Search and Exploration in the Tag Space," in *WWW 2006*, May 22–26, 2006, Edinburgh, United Kingdom.

ANALYSIS

A. Rejection of Claims 7 and 14 under 35 U.S.C. § 112, Second Paragraph

Appellants argue the claimed element, "wherein at least one of the two graph node clusters is empty, wherein an empty graph node cluster provides a structure for holding graph nodes at a future time," as recited in claims 7 and 14, is clear and unambiguous and, thus, claims 7 and 14 are not indefinite. *See* Appeal Br. 7. More specifically, Appellants argue the claimed "two graph node clusters" can be empty (i.e., not having graph nodes within them) and still can provide "a structure for holding graph nodes at a future time," as Appellants' specification states that, in one embodiment, a graph node cluster is both a null cluster (i.e., has no graph nodes within it) and capable of holding graph nodes in the future. *See* Reply Br. 2 (citing Spec. ¶ 47); *see also* Appeal Br. 7.

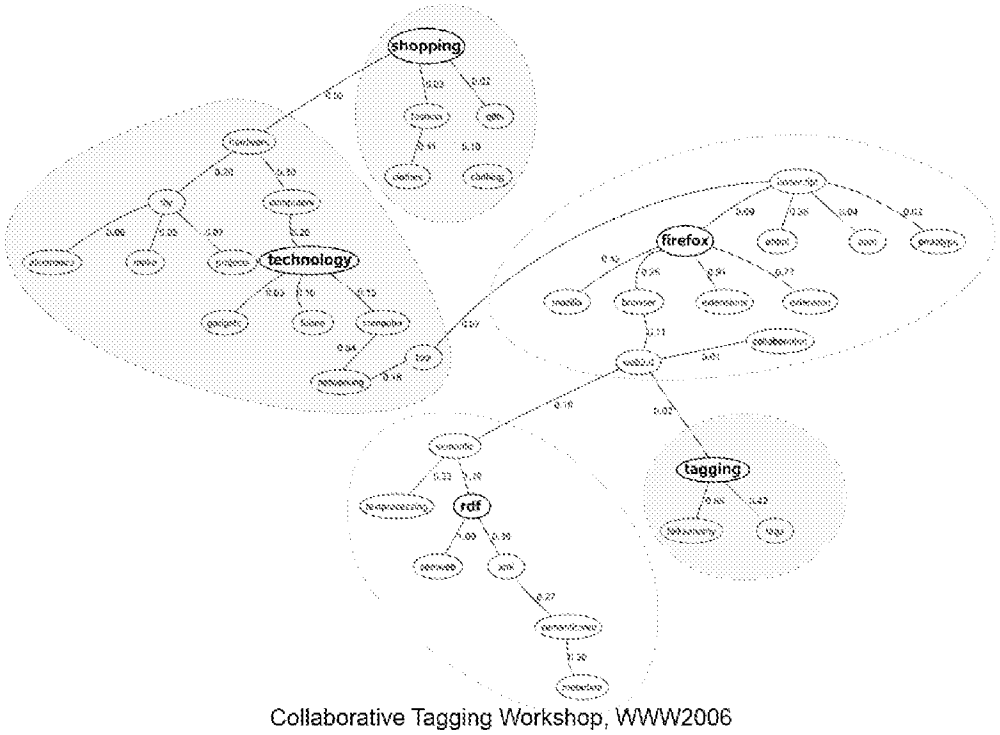
Appellants' argument is persuasive, and we agree with Appellants that claims 7 and 14 are not indefinite. The Examiner finds independent claims 1 and 8 recite "defining . . . multiple graph node clusters from the graph nodes," but also finds claims 7 and 14 further recite an "empty graph node cluster," and the Examiner further finds if a graph node cluster is empty it could not have been defined by the graph nodes. *See* Ans. 14–15. We disagree with this finding, as Appellants' specification shows an example embodiment where graph node clusters 506, 508, 510, 512, and 522 are defined based on graph nodes 504a–504j, and graph node cluster 522 is defined so as not to include any of graph nodes 504a–504j (i.e., graph node cluster 522 is defined as an empty graph node cluster), but is also defined to be capable of including graph nodes in the future. *See* Spec. ¶¶ 47, 49. A

person of ordinary skill in the relevant art would understand graph node cluster 522 to be defined based on graph nodes 504a–504j, even though graph node cluster does not include any of nodes 504a–504j. Accordingly, we do not sustain the Examiner's rejection of claims 7 and 14 under 35 U.S.C. § 112, second paragraph.

B. Rejection of Claims 1, 3, and 7 under 35 U.S.C. § 102(b)

Appellants argue Begelman fails to teach or suggest a cluster edge that describes a relationship between graph node clusters because the figure illustrated at page 17 of Begelman merely shows a node graph in which lines connect nodes to one another, rather than clusters. *See* Appeal Br. 8 (citing Begelman p. 17). Appellants further argue independent claim 1 recites "a cluster edge between two graph node clusters," and thus the edge defines a relationship between the entire two clusters, not just two nodes from the clusters. *See* Reply Br. 2.

The figure illustrated at page 17 of Begelman is reproduced below:



Collaborative Tagging Workshop, WWW2006

The reproduced figure depicts an overall cluster of related tags that are split into smaller individual clusters.

We do not find Appellants' argument persuasive. Although Appellants' specification defines a node edge as an edge that describes a relationship between two graph nodes and further defines a cluster edge as an edge that describes a relationship between two graph node clusters (*see* Spec. ¶ 47), those definitions do not prevent an edge from being both a node edge and a cluster edge when the edge connects two nodes of two different clusters, as the edge describes both a relationship between the two nodes and a relationship between the two clusters. Thus, we agree with the Examiner that an edge illustrated in the cited portion of Begelman that connects two nodes of two different clusters (e.g., an edge that connects the "shopping" node with the "hardware" node) teaches the claimed "cluster edge." *See*

Ans. 15–16. Further, we agree with the Examiner that an edge illustrated in Begelman that connects two nodes of two different clusters further describes the existence of a direct relationship between the two clusters. *See* Ans. 16. Therefore, we agree with the Examiner Begelman teaches "generating . . . a cluster edge between two graph node clusters from the multiple graph node clusters in the non-hierarchical graph database, wherein the cluster edge describes a relationship between the two graph node clusters," as recited in independent claim 1.

Because we are not persuaded that the Examiner erred in finding Begelman teaches all the elements of independent claim 1, we sustain the Examiner's rejection of independent claim 1 under 35 U.S.C. § 102(b). We further sustain the rejection of dependent claims 3 and 7, not argued separately. *See* Appeal Br. 8.

C. Rejection of Claim 2 under 35 U.S.C. § 102(b)

Appellants argue Begelman does not teach or suggest a cluster edge, or details of a relationship between two graph node clusters. *See* Appeal Br. 9; *see also* Reply Br. 2. Essentially, Appellants' argument regarding claim 2 is identical to Appellants' argument regarding claim 1. Appellants' argument is not persuasive for the reasons described above. Accordingly, we sustain the Examiner's rejection of claim 2 under 35 U.S.C. § 102(b).

D. Rejection of Claims 4–6 and 11–13 under 35 U.S.C. § 102(b)

Appellants argue Begelman fails to teach graph nodes that represent a "synthetic context-based object," where a synthetic context-based object is generated by "associating . . . a non-contextual data object with a context

object," as recited in claims 4 and 11.[3] *See* Appeal Br. 10. More specifically, Appellants argue that the Examiner failed to establish that any of the nodes illustrated at page 17 of Begelman (e.g., the "firefox" node or the "browser" node) represents a "synthetic context-based object," as recited in claims 4 and 11. *See id.* (citing Begelman p. 17). As argued by Appellants, the claimed "synthetic context-based object" is defined by at least one non-contextual data object and at least one context object, whereas the Examiner's analysis of Begelman's nodes in the Final Office Action merely identifies each node as either a "non-contextual data object" or a "context object," rather than the claimed "synthetic context-based object." *See id.*; *see also* Reply Br. 3.

We find Appellants' arguments persuasive. We disagree with the Examiner's finding that the specification fails to define the term "synthetic context-base object." *See* Ans. 19. Instead, as Appellants note, the specification indicates a "synthetic context-based object" is defined by at least one non-contextual data object and at least one context object, where the non-contextual data object ambiguously relates to multiple subject-matters, and where the context object provides a context that identifies a specific subject-matter, from the multiple subject-matters, of the non-contextual data object. *See* Reply Br. 3 (citing Spec. ¶ 54). We also disagree with the Examiner's finding that Begelman teaches graph nodes that include both a non-contextual object component and a context object component. *See* Final Act. 7. Instead, Begelman merely teaches nodes that

[3] Appellants also separately dispute the rejection of claims 5, 6, 12, and 13. *See* Appeal Br. 11–12; *see also* Reply Br. 3. We do not reach these arguments because the identified issue is dispositive, as claims 5 and 6 depend upon claim 4, and claims 12 and 13 depend upon claim 11.

represent related tags with edges between the nodes that identify a degree of similarity between the tags represented by the nodes. *See* Begelman, 17. While the connected nodes provide contextual information for each other (e.g., the "browser node" provides context for the "firefox" node), none of the nodes includes both a non-contextual object component and a context object component, and, thus, none of the nodes teaches the claimed "synthetic context-based object."

Accordingly, we do not sustain the Examiner's rejection of claims 4 and 11 under 35 U.S.C. § 102(b). We also do not sustain the Examiner's rejection of claims 5, 6, 12, and 13, which depend upon either claim 4 or 11.

E. Rejection of Claims 8–10 and 14 under 35 U.S.C. § 102(b)

Appellants argue Begelman fails to describe cluster edges that include upstream or downstream connections to nodes, much less clusters. *See* Appeal Br. 13; *see also* Reply Br. 4. Appellants' argument is persuasive, as the Examiner has failed to establish that Begelman teaches or suggests cluster edges that include upstream or downstream connections to clusters. While we agree with the Examiner that Begelman illustrates either an upstream or downstream connection from a node cluster to either an upstream or downstream node cluster (*see* Ans. 24), Begelman fails to teach or suggest the edges that connect the two nodes of the two node clusters include a description of the upstream or downstream connection.

Accordingly, we do not sustain the Examiner's rejection of independent claim 8 under 35 U.S.C. § 102(b). We also do not sustain the Examiner's rejection of dependent claims 9, 10, and 14 for the same reason.

F. Rejection of Claims 15–20 under 35 U.S.C. § 102(b)

Appellants argue Begelman fails to teach graph nodes that represent a "synthetic context-based object," where a synthetic context-based object is generated by "associat[ing] a non-contextual data object with a context object," as recited in claim 15.[4] *See* Appeal Br. 14. We find Appellants' argument persuasive for the reasons discussed above with respect to claims 4 and 11. Accordingly, we do not sustain the Examiner's rejection of independent claim 15 under 35 U.S.C. § 102(b). For the same reason, we also do not sustain the Examiner's rejection of dependent claims 16–20.

NEW GROUND OF REJECTION UNDER 37 C.F.R. § 41.50(b)

Claims 1–20 are rejected under 35 U.S.C. § 101 as failing to recite patent-eligible subject matter.

In *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, the Supreme Court articulated the required analysis for distinguishing patents that claim laws of nature, natural phenomena, and abstract ideas (which are not patent eligible under 35 U.S.C. § 101) from patents that claim patent-eligible applications of these concepts. *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2355 (2014). The first step in the analysis is to determine whether the

[4] Appellants also separately argue that Begelman fails to teach cluster edges that connect clusters, or instructions that "combine synthetic context-based objects that each contain the same non-contextual data object and a different context object from a second set of different context objects into a second synthetic context-based objects graph node cluster." *See* Appeal Br. 14; *see also* Reply Br. 4. We do not reach these arguments because the identified issue is dispositive. Further, Appellants' argument regarding cluster edges has also previously been addressed above with respect to independent claim 1.

claims at issue are directed to a patent-ineligible concept, such as an abstract idea. *Id.* If so, the second step in the analysis is to consider the elements of the claims individually and as an ordered combination to determine whether the additional elements transform the nature of the claim into a patent-eligible application. *Id.*

With respect to the first step, we conclude that independent claims 1, 8, and 15 are directed to the abstract idea of correlating related information. *See Digitech Image Technologies, LLC v. Electronics for Imaging, Inc.*, 758 F.3d 1344, 1350–51 (Fed. Cir. 2014) (holding that "a process of organizing information through mathematical correlations . . . not tied to a specific structure or machine" is an abstract idea). Similar to the claim at issue in *Digitech*, independent claims 1, 8, and 15 recite a process of correlating related information—i.e., organizing graph nodes into multiple graph node clusters and cluster edges connecting the multiple graph node clusters. Thus, we conclude that independent claims 1, 8, and 15 are directed to abstract ideas.

With respect to the second step in the analysis, we see nothing in independent claims 1, 8, or 15 that would transform the patent-ineligible concept of organizing information into a patent-eligible concept. Independent claim 1 recites "receiving, by the processor, a data stream that describes graph nodes in a non-hierarchical graph database," "defining, by the processor, multiple graph node clusters from the graph nodes in the non-hierarchical graph database," and "generating, by the processor, a cluster edge between two graph node clusters." Such well-understood, conventional data processing steps "amount to nothing significantly more than an instruction to apply the abstract idea . . . using some unspecified, generic

computer," and thus are "not enough to transform an abstract idea into a patent-eligible invention." *See Alice*, 134 S. Ct. at 2360 (internal quotation marks and emphasis omitted). Because the generic hardware components recited in independent claims 8 and 15 (i.e., "a tangible non-transitory computer readable storage medium" and "computer system") also fail to meaningfully limit the claimed abstract idea beyond "implementation via computers," those claims fail under *Alice* for substantially the same reasons. *See id.*

We have also reviewed dependent claims 2–7, 9–14, and 16–20, and we do not see anything in those claims that transforms the patent-ineligible concept of organizing information to a patent-eligible concept. Accordingly, we find that claims 1–20 fail to recite patent-eligible subject matter under 35 U.S.C. § 101.

37 C.F.R. § 41.50(b) provides that "[a] new ground of rejection pursuant to this paragraph shall not be considered final for judicial review." 37 C.F.R. § 41.50(b) also provides that Appellants, *WITHIN TWO MONTHS FROM THE DATE OF THE DECISION*, must exercise one of the following two options with respect to the new ground of rejection to avoid termination of the appeal as to the rejected claims:

(1) *Reopen prosecution.* Submit an appropriate amendment of the claims so rejected or new evidence relating to the claims so rejected, or both, and have the matter reconsidered by the Examiner, in which event the proceeding will be remanded to the Examiner . . . .

(2) *Request rehearing.* Request that the proceeding be reheard under § 41.52 by the Board upon the same record . . . .

DECISION

We reverse the Examiner's rejection of claims 7 and 14 under 35 U.S.C. § 112, second paragraph.

We affirm the Examiner's rejection of claims 1–3, and 7 under 35 U.S.C. § 102(b).

We reverse the Examiner's rejection of claims 4–6 and 8–20 under 35 U.S.C. § 102(b).

We newly reject claims 1–20 under 35 U.S.C. § 101.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv)

AFFIRMED-IN-PART
37 C.F.R. § 41.50(b)